In Actas de *Interacção 2006 - 2ª Conferência Nacional de Interacção Pessoa-Máquina*, Braga, 16-18 Oct. 2006, GPCG Editor, ISBN: 972-98464-7-2, pp.123-131.

Engaging Virtual Agents

Ido A Iurgel
Dep. Digital Storytelling, ZGDV e.V.
Rundeturmstr. 10, 6423 Darmstadt, Germany
Ido.Iurgel@zgdv.de

Adérito F Marcos
Departamento de Sistemas de Informação, DSI
Universidade do Minho, Portugal
marcos@dsi.uminho.pt

Abstract

Embodied virtual assistants normally don't engage the user emotionally. They fulfil their functions, e.g. as shopping assistants or virtual teachers, factually and emotionless. This way, they do not explore the full potential of the presence of an embodied character. In real life, the personality of the teacher or salesperson, their ability to involve and even to entertain is essential for their success. But how much of these "soft factors" can be translated into behaviour of virtual agents? Which kinds of virtual personalities are appropriate for which group, and in which context?

We call virtual agents with engaging "soft skills" Engaging Virtual Agents. This paper presents a software platform employed for experimenting with soft skills and for creating different personalities of virtual agents. The focus of this platform is on authoring principles that facilitate the cooperation of content creators and computer scientists. We also present "Julie", an example that was shortly concluded as part of a research project commissioned by SAP AG. Julie is a virtual sales assistant that employs actively emotional expressions and narrative techniques, in order to provide additional motivation for the customer to visit and to remain at the virtual shop.

Key-words

Embodied Virtual Agent, Virtual Character, Authoring, Personality, Emotion, Virtual Assistants

1. INTRODUCTION

In daily life, interacting with a person that is fulfilling a certain duty – a functionary or employee, a sales person, a teacher, a work colleague – normally involves more than purely accomplishing rationally "hard" transactions. "Soft", emotional factors play an eminent role for the success and the stability of the processes. E.g., a teacher should expect better learning results and continuity of work if the students feel at easy with him or her; a sales person is more successful if the customer feels sympathy and laughs at the jokes of the vendor. In short, "soft", social and emotional skills matter very much, in daily life. Their importance for the efficient operating of organizations and for the accomplishment of rational tasks in a social context is well known and studied.

We employ the term "virtual assistant" in a wide sense that covers any kind of virtual functionary, e.g. teachers or sales persons. Currently available embodied virtual assistants show only very rudimentary soft skills and personality traits. E.g., a virtual shopping assistant usually displays a constant smile as the only sign of emotional involvement, remaining otherwise passive and emotionally shallow.

We think that a great potential of engaging the user is being missed, if virtual functionaries are not endowed with advanced soft skills. But what exactly are the soft skills required for a technically feasible virtual agent? Which kind of emotional, not task oriented behaviour is useful, and what is superfluous or even disadvantageous? Certainly, R&D on social skills for virtual agents has to start with everyday phenomena of human-human interactions; but there are many reasons not to aim at an exact simulation of human soft skills.

It is important to keep in mind that a virtual agent remains a metaphor; there are certain analogies between a virtual and a human agent, but there are also essential differences. First, we won't succeed in simulating exactly human social behaviour in virtual agents, in any predicable future, because of technological limitations. E.g., virtual agents are far from being able to understand subleties of emotional and verbal expression of a human counterpart, and thus cannot be very precise when employing social skills. We must take those technological limitations into account, when designing social skills for virtual agents; otherwise, we would postpone really building them into an indefinite future.

Figure 1: Anna, the shopping assistant of IKEA, is an example of a monotonous virtual agent (www.ikea.com)

Second, social skills for virtual agents must differ from human skills because virtual agents are perceived and evaluated differently by end users [Bente02]. For example, a charming smile of a virtual assistant won't touch a human user as deeply as the smile of a sales person; but a behaviour that would be incontestable for a real functionary, e.g. lamenting about an unhappy love affair, might be funny and entertaining when presented by a virtual agent.

Because the required social skills and personality traits of virtual agents are not yet well understood, we have build a software platform for virtual agents that facilitates the creation of different virtual agents with varying soft skills. We think that content creators require tools and methods to experiment with different ideas and intuitions, thus incrementally developing technically feasible and appealing virtual behaviours. It is not accurate to implement models derived from the analysis of human behaviour without taking into account the peculiarities of the required behaviour of virtual assistants. For example, virtual agents' "emotion psychology" might be required to be rather different from that of humans, because of the specific perception of virtual humans by end-users. Therefore, it is necessary to allow content creators to experiment with ad-hoc models and behaviour structures, and currently, the major R&D task consists in developing a framework that enables these experiments, in cooperation of content and of computer science specialists.

In this paper, we will discuss the software platform employed to experiment with social skills and personalities of virtual assistants, and we present *Julie*, a virtual shopping assistant created as an example for SAP AG. After a review of related work in section 2, we will discuss, in section 3, some important general problems of creating soft skills for virtual agents. Section 4 will expose the technical platform, with emphasis on authoring issues, and section 5 describes the exemplary implementation of *Julie*. Section 6 addresses future issues, and section 7 contains concluding remarks.

2. RELATED WORK

A. Gulz [Gulz05] provides an excellent overview and discussion of social skills in the realm of virtual assistants, especially concerning virtual tutors. Though some very sophisticated systems were developed (e.g. Bickmore [Bickmore03], Baylor and Kim [BaylorKim05]), no system was build from scratch on with emphasis on efficient authoring and on fostering experiments by content creators. Gebhard et al. [Gebhard03] have developed an authoring system for interactive applications; their solution does not integrate chatting and does not allow dismissing the edges of the directed graph they employ.

Astonishingly, none of the agents was build with the expressed intention of being entertaining and funny, and no behaviour patterns inspired by drama were employed. Nijholt [Nijholt05] has discussed humorous acts of embodied virtual agents to smoothen human-computer interaction problems. Most current virtual shopping assistants are commercial products, and their details are not published. However, no one seems to possess the guiding and pro-active faculties presented in this paper. The technology discussed in this paper is based on the work described in [IurgelZiegler05].

3. SOFT FACTORS FOR VIRTUAL AGENTS

Basically, we are currently "decorating" minimal, functional behaviour of virtual agents with entertaining, relaxing, or motivating additional elements. Therefore, the soft skills of the agents must be interwoven with rational and functional faculties. The concepts and technologies for interweaving assistance functions and soft factors are at the focus of the research presented in this paper. We want to provide a framework for experiments that will also help us understanding what assisting virtual characters can sensibly do, besides fulfilling their duties.

At this stage, we are interested in short remarks and short emotional dialogue games that the virtual agent can throw in whilst providing information, and in short social games that enclose the functional, assisting behaviour. I.e., we are studying how to maintain the assisting functions while at the same time adding more social and emotional factors. For example, we want a virtual shopping assistant that always will say the price of an item, when the user requests it, thus staying within its role, but that also throws in emotional remarks and interjections, and that plays engaging "mini-dramas".

Mini-dramas are temporal structures that create suspense and end with a resolution. A mini-drama is a minimally short narrative structure[1]. It aims at providing additional structuring of time for the user, who should feel compelled to learn the solution. We expect mini-dramas to enhance motivation, enjoyment and fun. But we also expect them to enhance the social and emotional dimension of the communication, in an analogy to the creation of social bounds and reciprocity in real human interaction.

The suspense of a mini-drama should provide additional reasons for the customer to remain at the shopping site, or for the pupil to continue his e-learning session. E.g., Julie (cf. bellow) will begin throwing in unhappy and nasty remarks which she[2] first refuses to comment (conflict),

[1] A mini-drama is roughly equivalent to what Mateas and Stern [MateasStern05] call a beat, following the terminology of McKee [McKee97] though the term beat wouldn't be appropriate here because a mini-drama is not necessarily embedded within a larger narrative structure, but can stand on its own.

[2] We think that is sounds more natural to employ the personal pronoun "she", instead of "it", when referring to the female name "Julie".

increase their intensity until a climax is reached, and then confess that she desires to become friend with the user, but doubts whether this wish is reciprocal (resolution). We expect the users to be pleased and laugh at this turn around, and not to be deeply touched by this virtual advance. Following drama theory [McKee97], this mini-drama has revealed a hidden, unexpected trait of Julie's "personality", namely her (alleged) affection towards the user; this uncovering of a "hidden", unexpected layer is exactly the main element of a good story. As in cliff-hangers of TV-soaps, mini-dramas could also be used to create continuity, suspense and curiosity when returning to a learning session or to an internet shopping site.

Developing mini-dramas require much authoring efforts. Automated creation of mini-dramas out of general and generative drama rules is out of current technology's reach. This means that only a limited set of mini-dramas will be available, for any application. This poses challenges to commercial exploitation; there, a constant production of additional mini-dramas for regular customers might be necessary.

It is not common for real sales persons or teachers to actively employ mini-dramas to engage their counterpart. But this is an example of the major difference between virtual and real agents, and clarifies why we need a framework for experimenting with the specifics of virtual assistants: A real teacher could loose his/her face and privacy, when uncovering hidden personality layers and feelings; a real sales person could strategically flirt with a customer, but would appear aggressive and loose credibility when playing with deep feelings. Those are no concerns, when it comes to a virtual teacher or shopping assistant.

The really challenging interaction design questions in the virtual realm are related to balancing and estimation of effect. "Balancing" means that, in applications that employ virtual assistants, some hard criterion for success exist and that extra-functional behaviour of the virtual agent must, at the end of the day, have contributed to this goal. As in any real working context, we are better of with social skills, but we won't achieve the targets if we only focus on socializing. Some difficult balancing is required. For example, the funny behaviour of a virtual shopping assistant must not demand the whole attention of the customer, but should be useful in attracting more site-visitors and motivate them to buy. A virtual tutor in a business context must not consummate too much time with the creation of emotions, because the emotions must be beneficial for the final learning success.

"Effect" is a problematic design aspect because the impact on a particular user is still very difficult to foresee. Here, again, the analogy between a virtual and a real human can be misleading. If the user laughs at a joke told by a virtual character, how much is he/she laughing about the joke, and how much about the fact that a soulless thing is pretending to tell a joke, employing a dry synthetic voice and imprecise mimic? In the later case, would the user still laugh at the n-th joke? The effect of a behaviour well known on humans is not easy to foresee, when simulated by a virtual human.

Though balance and effect can be evaluated formally, the process of optimising them is essentially an intuitive, experience driven, and even artistic process, in the same vain as screenwriting. Hence the importance of a system that supports experiments and that can be authored by content creators that are not computer specialists.

4. THE SOFTWARE PLATFORM

4.1 Implementation

The runtime environment is a client-server system for the internet which supports virtual characters that are

- proactive
- assisting and guiding
- able to throw in sensibly remarks, jokes, teasing, and other interjections
- able to play mini-dramas
- responsive to mouse click and able to chat
- flexible and easy to author

The server and the authoring tool were written in Java, the communication employs AJAX[3]. The complete logic is on server side; on the client side, we use the internet explorer with the Haptek[4] plug-in for virtual character rendering, which employs the speech synthesis of the Windows client through the Microsoft SAPI 5.1 API. The authoring tool *Cyranus* was developed at ZGDV e.V. (cf. [Iurgel06]), using Java Swing.

4.2 The Quest for Accessible Authoring

Accessibility of authoring is an essential requirement for the envisaged target of intuitively experimenting with possible social skills for virtual assisting agents. Content creators with some computer affinity, but who are not computer scientists, must be able to use the tools. Equally, the authoring system must be easily adaptable and customizable by computer scientists, in dialogue with the content creators.

The most established principles for authoring of interactive applications are some variation of a directed graph. They are very intuitive when the graphs are not too dense, the guard conditions not complicated, and the amount of possible paths not too large. In the scenarios we envisage, there are many situations in which directed graphs can be used. Thus, an important feature of our approach consists in enabling the use of directed graphs wherever the content creator deems it appropriated, and of other control strategies when necessary. There must be a smooth transition from the use of directed graphs to those other technologies, and the author must be able to begin with graphs, or even with a linear sketch, and be

[3] "Asynchronous JavaScript Technology and XML", a technology that extends the use of Javascript within web applications.

[4] Cf. http://www.haptek.com/

able to later replace the sketch easily, when the graph has become too complex.

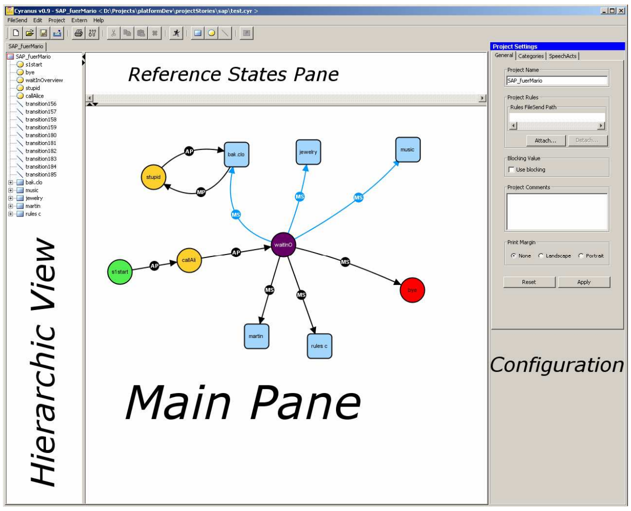

Figure 2: Screenshot of *Cyranus*. On the left a tree view of the hierarchy, on the right the configuration panels

Short dialogue games are an example of the sensible use of directed graphs within an interactive application with natural language (typed) input. Of course, only short pieces of the dialogue are predictable enough to be handled by graphs.

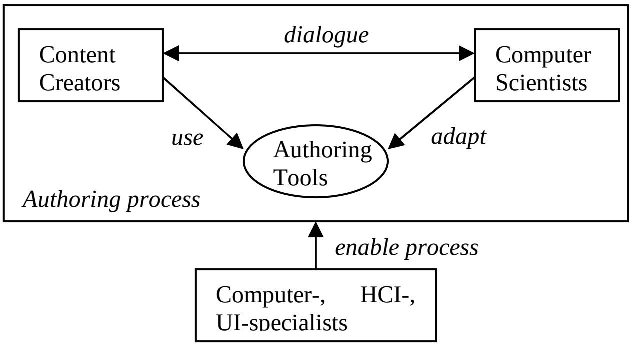

Figure 3: The process of adaptation of the authoring tools by computer specialists must be already addressed when designing the authoring tools.

For these cases we allow, at any time, for the use of a hierarchic transition network, inspired by Harel's Sate Charts [Harel88]. Figure 2 is a screenshot of the authoring tool *Cyranus*. The figure depicts the directed graph representation, derived from State Charts. Composite states are represented by squares; double-clicking on a square will show the states that are encapsulated by the composite state, i.e. the state that are "inside the square". Terminal states are circles; circles do not contain other states inside.

It is well known that directed graphs cannot be employed universally; for example, they are inappropriate if a large number of permutations over a set of states is possible. Figure 4 shows an already forbiddingly complex graph over a set of only six states of *Cyranus*; the most complex guard conditions required to control the flux are not shown. Because of this complexity, other "next state" activation procedures must be devised.

Other control strategies than the use of directed graphs are integrated by a delegation chain: An external event that does not lead to the activation of a state within the transition network framework is passed over sequentially to other so-called "activation engines", which in their turn could activate a state, in this case breaking the chain of delegation. The engines are also permitted to fire without the occurrence of an event. Every activation engine is attached to a single composite state, through meta-data of this state, and it may activate only terminal or composite states that are located within this composite state. Thus, encapsulation is fully preserved.

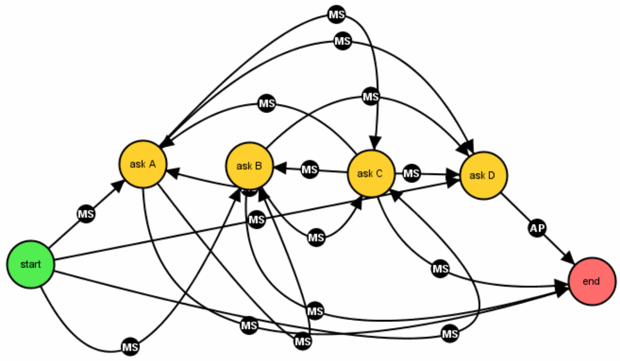

Figure 4: A scenario where different inputs may cause many permutations over a set of states

We currently employ two activation engines: The production system Jess[5] (we are up to now employing solely the forward chaining rules), and a similarity based engine that shall implement principles of CBR (Case-Based Reasoning), which is currently being developed within the German *Virtual Human* project[6]. As a fall back engine, we also employ the chatterbot *Alice*[7]. This is not an activation engine, because it does not activate a state that is within reach of all activation engines. Instead, the *Alice Engine* generates autonomously an action that is send to the client. *Alice* is employed here because of the huge data bases that can be freely downloaded; this move extends the chatting capabilities of the virtual agents. Figure 6 represents the delegation chain; in the example of this figure, the CBR Engine (Case-Based Reasoning) only is able to handle the external event, by activating a state of the composite state this engine is attached to.

Any state that the engines of the activation chain may activate is represented by Cyranus as either a square or a circle. I.e. every sate that could be activated is visually

[5] Cf. http://herzberg.ca.sandia.gov/jess/

[6] Cf. www.virtual-human.org

[7] Cf. www.alicebot.org

represented, thus guaranteeing a high degree of visual control of the possible actions of the system (with the exception of the fallback engine for ALICE, cf. above).

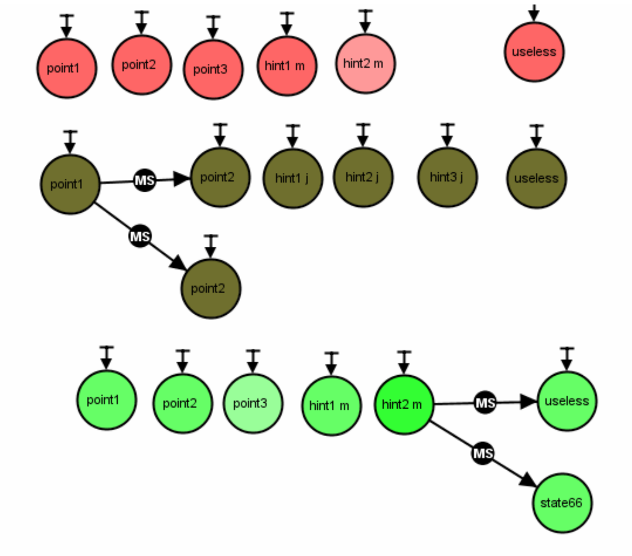

Figure 5 : Employing the transition network together with activation engines

States can be activated by activation engines because they carry meta-data. We expect content creators to be able to use the directed graphs, with the help of simplifying GUIs for specifying guard conditions, and to describe the meta-data of the states that an activation engine shall be able to activate, equally employing GUIs. We normally do not expect an author do express formal rules or new similarity functions. This is the work of computer scientists, which will work in dialogue with the content experts.

Figure 5 depicts the combined use of activation engines and of the transition network. Note that most of the circles – terminal states – are not attached to transitions, and that they have an arrow on top. This arrow symbolizes that they contain metadata, and can thus be activated by an activation engine. Note also that it is possible to combine transitions with the activation engine. Only a single state can be active at a given moment, parallelised activation is not yet being supported. If an arrow connects the currently active state to another state, in the visual representation of *Cyranus*, the system will first check whether this transition is passable; in case that the guard conditions do not hold (transition not passable), the delegation chain (cf. bellow, and figure 6) will result in the calling of subsequent engines, which may activate any state which contains appropriate meta-data – on the representation of *Cyranus*, this is any state (square or circle) that shows a short symbolic arrow on top. The activation engines are adapted by computer specialists for producing the required behaviour, and a content specialist then only has to add states and employ the meta-data to take profit of the engines, without knowledge of any implementation detail.

Once a rule set or function is created, it can be reused with new content. This dedicated process of adaptation and enhancement (fig. 3) is an important new feature of the authoring system: We should not expect work on the authoring tools to be concluded, at the beginning of a production, since content creators might always come out with new ideas and requirements; the tools therefore should provide dedicated extension procedures that efficiently end up with the required features, whilst still remaining accessible for the content creators. This considerable extends the reach of the authoring process.

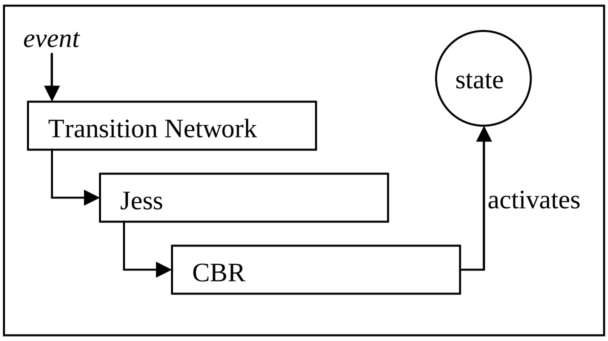

Figure 6: The chain of delegation of events to different engines. In this example, the CBR-engine responds

Thus, our authoring tool *Cyranus* also offers neat APIs and configuration for adapting the GUIs that are used to create meta-data, guard conditions, and action expressions. Internally (i.e., not visible for the content specialist), they are described with generically expressive XML (e.g., the guard-conditions are kept internally as XEXPR[8], a very powerful, general, functional programming language in XML). For example, figure 7 shows partially the GUI for scripting utterances and other behaviours of the virtual agents. This exchangeable scripting GUI is integrated within *Cyranus*, and it outputs XML to the main *Cyranus* framework, which integrates this XML into possibly more complex XEXPR expressions. The non-expert content creator will not be involved into any XML-coding issues.

5. JULIE – AN EXAMPLE OF AN ENGAGING VIRTUAL AGENT

We've created an exemplary virtual shopping assistant for SAP AG, in order to study both the creation process of virtual assistants and certain non-standard behaviour patterns for virtual assistants.

[8] Cf. http://www.w3.org/TR/xexpr/

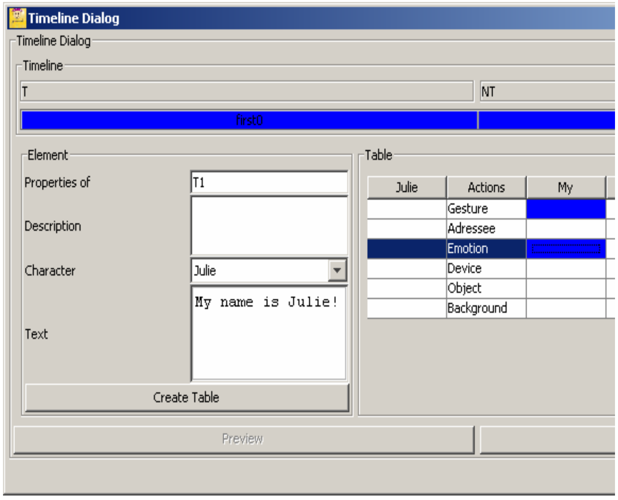

Figure 7: Part of the GUI for scripting behaviours for virtual agents

5.1 Julie's Concept

The exemplary cloths shopping site is intended for young women and girls. We expect that a considerable part of this group will react to attributes as "cool", "trendy", "funny". The visual design of the site is inspired by real high class cloth shops, in order to create a young, sophisticated and demanding atmosphere that differs from usual internet shopping sites. Figure 8 exemplifies the design of the shopping site. It depicts the main page of the cloth section. The user clicks on one of the cloths displayed on the right, and the item moves forward, as in a catwalk. The user may chat with Julie at any time, using the chat-box located bellow Julie.

We wanted the shopping assistant *Julie* to be able to guide the customer through her shopping experience. Julie should comment on mouse clicks of the user, e.g. saying "If I were not caged in this computer, I would myself by this jacket!", or initiating short dialogue games, e.g. "do you want me to tell what I really think about this cloth?" – "yes, please" (typed) – "it will make you look overweight". Julie should not be repetitive when the user clicks more than once on an item, and should actively guide the user to items not yet examined, e.g. saying "look at the jacket on the top right corner. It will certainly fit you!" – (user clicks at item) – "And? What do you think? Am I not right?"

Julie's remarks are not only informative, but sometimes teasing, joking or even bitchy. Of course, this is intended to create fun. Julie is pro-active, and will e.g. sometimes try to instigate some off-topic chat, e.g. on the party of last night. Julie also can comment on the overall session and the reactivity of the user, e.g. saying "you are wasting my time if you do not do anything". At any time, the user can freely click on the items and chat with Julie.

These examples already indicate the personality profile we have created: Julie is much more active, demanding, direct, and even nasty and aggressive than other virtual assistants.

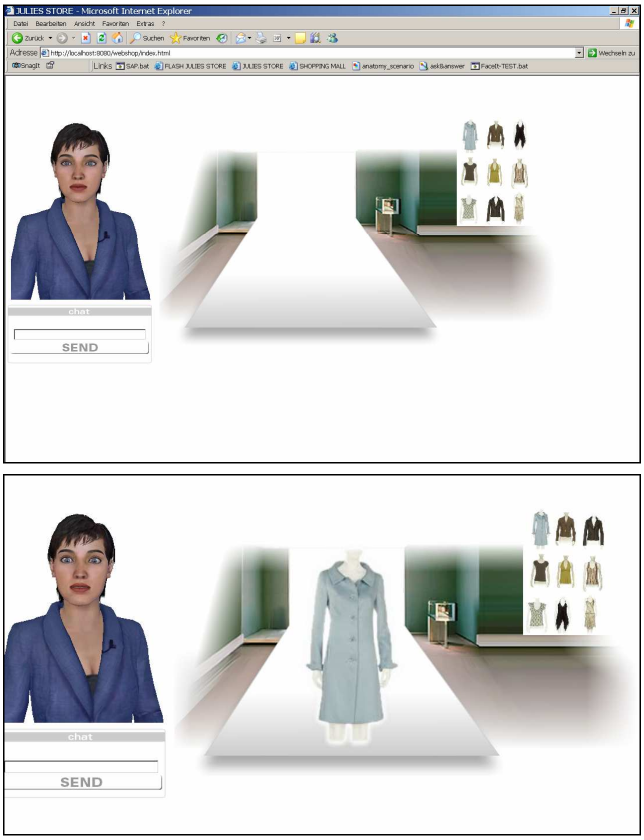

Figure 8: The design of the shopping site for *Julie*

Julie is intended to be a rather extreme example of a personality rich, active, emotional assisting agent. We want to understand whether and by whom this is perceived as distracting and offensive, or funny and attracting. For commercial use, marketing considerations would certainly require the reduction of some parameters.

We are also experimenting with mini-dramas for Julie. Currently, we have implemented the mini-drama "I want to be your friend". Julie will begin by throwing in some nasty remarks like

"I don't think that we are going to get along with each other" – "why?" (typed) – "I do not want to tell".

A bit later, Julie will resume the topic:

"Do you really want me to tell you what is the problem between us?" – "yes" (typed) – "I think I really want to be your friend, but this is not possible…" (makes a very sad face).

This mini-drama shall, at its beginning, create suspense, because the user shall be intrigued to know Julie's reasons. It shall after its resolution initiate some more intimate exchange, and the creation of some sort of social bounds between user and Julie.

5.2 Technological Issues

The guiding behaviour is generated by a rule-set expressed in Jess. Figure 9 is a short example of a rule written in Jess. Jess activates a corresponding state, if the user clicks on or mentions a piece of cloth, and activates states that guide the user to items not yet examined.

```
(defrule updateNextPoint "keeps priority for click
states up to date"
  (declare (salience 100))
    (state (id ?id) (visits ?v))
    (meta (state ?id) (name hotspot) (value
?region))
    (meta (state ?id) (name priority) (value ?p))
    (meta (state ?id) (name type) (value point))
    (start (id ?id) (visits ?v))


    ?next <- (next (hotspot ?region) (type point)
(priority ?oldp&:(< ?p ?oldp)))


    ?repeat <- (repeat (hotspot ?region))
```

Figure 9: Example of a Jess-rule

The states carry meta-data as

```
<ruleData hotspot="partyDress" priority="5" type="point"/>,
```

indicating that this state shall be activated by the Jess-engine whenever the user clicks for the fifth time on the party dress item.

The meta-data

```
<ruleData hotspot="partyDress" priority="1" type="hint"/>,
```

for example, means that this state will be activated only if the party dress is not being noticed at all, and that the state contains behaviour that will guide the user to look at this dress.

States can also be activated by typed user input. For the resolution of the input, we employ either a regular expression, or reuse the so called “reduction” mechanism provide by *Alice*. E.g., the user input

> “do you love the grey coat?”

will be changed internally into

> “do you like the grey coat?”.

This can be matched against the meta-data

```
<ruleData hotspot="partyDress" speechact="do you like the grey coat?" priority="1" type="chat"/>,
```

and then the state can be activated. Note that a short dialogue game, authored with directed graphs, could follow:

> “do you like the grey coat?” (typed) – “yes, very much. What’s about you?” – “I don’t know” (typed) – “if I were you, I would buy it!”

The guard conditions for the transitions of this short dialogue game can contain either regular expressions to catch up predictable input, represented internally for example as

> (\w|\s|\d)*yes(\w|\s|\d)*

for any sentence containing “yes”, or again recur to the reduction functions of *Alice*.

Actions of states are mostly dialogue acts expressed in XML, which are send as direction to the client. The format is derived from RRL [Piwek02]. For example, the sentence “don’t you feel good?” will be coded as

```
<dialogAct>
 <sentence id="s_1147866678280">
  <word id="w0">Don't</word>
  <word id="w1">you</word>
  <word id="w2">feel</word>
  <word id="w3">good</word>
  <textSign id="ts0">?</textSign>
 </sentence>
</dialogAct>.
```

5.3 First Experiences

We have not made yet a formal evaluation of *Julie*. Our informally acquired feedback about Julie’s behaviour ranges between extremes. Seldom had we observed indifference. For example, a middle aged woman without computer science background felt sincerely repudiated and preferred not to be confronted twice to Julie. But more often, guests and occasional users laugh out loudly at Julie, and immediately start explorations of the system, in expectation for the next funny thing to happen. Experiences with the reaction of a group of young girls of the target age, visiting the German institute, indicated that we should be on the right track.

Concerning the authoring process, the experiences are very favourable. For example, many of the funny remarks and dialogue games of Julie were made by a 16-year old pupil without any deeper computer background who was making a short internship, and the total amount of time required for instructing him did not exceed some very few hours. *Cyranus* has proven to be flexible and powerful enough to be employed in practice, without structural changes, within three other projects.

We initially underestimated the learning curve of the rule-system Jess. It seems now out of reach that someone without a computer science background should extend the rules of the system, which we at the beginning thought would be possible.

6. FUTURE ISSUES

We are currently implementing a second experimental scenario for Julie as a kiosk within a shopping mall. The distinctive feature of this variation is the video-based emotion recognition of the user, and the integration of this input into the behaviour of Julie. In this scenario, Julie will explain the way to different shops of the mall, but will also, from time to time, tell a joke. We need the emotion recognition to detect whether the visitor enjoys Julie’s attempts at being funny. This is likely to vary very much among visitors. The frequency of entertaining remarks will depend on the recognized emotion. In this scenario, adaptation of the system to the user is important because the visitors cannot be expected to configure or to adapt to the system.

Within scenarios of longer duration, e.g. tutoring systems or certain internet shopping sites, it is certainly more important that the user is enabled to adapt the system to his/her preferences by choosing a preferred virtual character, than creating a virtual agent that would try to adapt to the very different possible user types.

We have not yet made any formal evaluation of the "Julie"-system, since we are currently still at the phase of acquiring experiences and fine-tuning content. In fact, a formal evaluation of Julie would certainly prove to be a most difficult endeavour. The evaluation of Julie would be a success if a shopping site sells more because of its social and emotional behaviour. This might be the case even if a majority of potential customers strongly dislike Julie, as long as a relevant minority enjoys this assistant.

Most probably, Julie will remain, for us, an in-between step useful for acquiring experiences, propagating ideas and fine-tuning technology. Only the next assistant is going to be formally evaluated, which is likely to be a virtual tutor, because the learning effects of a virtual tutor are easier to evaluate. In a former evaluation of a learning system that employs virtual characters[9], we have found some positive effects, and indicators of "pleasure" were significantly high. But, perhaps astonishingly, the pleasure was more associated with "relaxation" than with "excitement", which goes along with other findings about the fear reducing effects of virtual characters [Bente02]. An important question for future evaluation of our system will be whether it will be possible to maintain or increase positive effects, while moving the "pleasure" indicator to the side of "excitement". This is presumably important for example when younger user groups shall be addressed.

A major issue that is being tackled concerns augmenting the generativeness of the system. E.g. Julie employs shallow natural language processing and shallow emotional behaviour. I.e., no generic modules are employed that would allow Julie to generate dialogue acts and dialogue games out of language independent data bases and context information, and no emotional models control Julie's emotions. They are scripted, and unforeseen sequences of scripts may cause weird emotional behaviour. It was necessary to start shallowly because generated behaviour is not entertaining and witty. But on the long run, a combination of both scripted and modelled behaviours will be necessary, for the sake of modularity, parameterisation, and diminution of the authoring expenses.

7. CONCLUSION

We think that virtual embodied agents possess important potentials for motivating, relaxing, or entertaining users, when they present some social skills which they can employ without neglecting their proper functions as virtual teachers, sales persons, or servants. Following Gulz, we do not expect any simple, generally valid findings. Some people are likely to adore some kinds of virtual characters, others will hate them, and some persons will prefer to dispense with virtual agents altogether. The social preferences and needs for real or virtual company are certainly very different and individual. Engaging virtual agents also introduce a role play and game element into possibly serious contexts that won't be appreciated by everyone.

But nevertheless, many people are likely to profit from engaging virtual assistants, to enjoy them and to experience pleasure at their presence. The current practice of creating neutral, inoffensive virtual assistants with little personality is probably missing the point, because differences between preferences of different persons are being neglected, and, at the end, only very few persons really end up enjoying the virtual agent. A virtual personality that is offensive for a certain group can be very funny for another; an agent that evokes protecting and nurturing instincts (reminiscent of the Tamagotchi) won't captivate everybody, but certainly a relevant minority. We think that the solution lies in learning to create different, pronounced personalities of virtual assistants that will please certain groups, and to let the persons choose the character, if any, that is most appropriate.

The software platform that we've presented enables the creation of different virtual personalities with different social skills, in cooperation of content creators and computer scientists, and to experiment with these agents. *Julie*¸ the virtual character that we have presented as an example is rather active, and a bit nasty and aggressive, and employs emotionally loaded mini-dramas to catch the attention of the user. Many people enjoy and laugh at this.

REFERENCES

[BaylorKim05] A.L. Baylor, Y. Kim. Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education* 15, 1-18.

[Bente02] G. Bente, N. C. Krämer. *Virtual gestures: Analyzing social presence effects of computer-mediated and computer-generated nonverbal behaviour.* Paper presented at Fifth Annual International Workshop Presence 2002, 8.-10. October 2002, Porto, Portugal.

[Bickmore05] T. Bickmore. Relational agents: effecting change through human-computer relationships. PhD Thesis, MIT.

[Gebhard03] P. Gebhard, M. Kipp, M. Klesen, T. Rist. *Authoring Scenes for Adaptive, Interactive Performances.* In: Proc. of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03), Melbourne.

[9] This yet unpublished evaluation was conducted within the art-E-fact-project, cf. www.art-e-fact.org, IST-2001-37924. More information on this evaluation's results can be obtained from the authors upon request.

[Gulz05] A. Gulz. *Social enrichment by virtual characters – differential benefits.* Journal of Computer Assisted Learning 21, 405-418.

[Harel88] D. Harel. *On visual formalisms.* Communications of the ACM. V. 31 I5 514-530.

[IurgelZiegler05] I. Iurgel, and M. Ziegler. *Ask & Answer: An Educational Game where it Pays to Endear your Capricious Virtual Character.* IVA 05, International Symposium on Intelligent Virtual Agents, September 12-14, Kos, Greece.

[Iurgel06] I. A. Iurgel. *Cyranus - An Authoring Tool for Interactive Edutainment Applications*. In: Lecture Notes in Computer Science (LNCS), Springer, S. No. 3942 (2006), p. 577-580.

[MateasStern05] M. Mateas and A. Stern*. Procedural Authorship: A Case-Study of the Interactive Drama Façade.* Digital Arts and Culture (DAC), Copenhagen, November.

[Nijholt05] A. Nijholt. *Conversational agents, humorous act construction, and social intelligence.* In K. Dautenhahn (Ed.), Proceedings AISB 2005: Social Intelligence and Interaction in Animals, Robots and Agents. Symposium on Conversational Informatics for Supporting Social Intelligence and Interaction, 1–8. Hatfield, England: University of Hertfordshire.

[McKee97] R. McKee. Story: Structure, Style, and The Principles of Screenwriting. Regan Books.

[Piwek02] Piwek, P., Krenn, B. Schröder, M. Grice, M., Baumann, S. Pirker, H. *RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA.* Proceedings of the AAMAS workshop on "Embodied conversational agents - let's specify and evaluate them!", Bologna, Italy.